Appl. No. 10/625,334
Amendment and Response dated February 16, 2007
Reply to Office Action of January 29, 2007

Amendments to the Specification:

Please replace paragraph [0026] with the following rewritten paragraph:

At 102, a dominator tree may be created for one or more basic blocks in a program. At 104, ϕ-nodes may be inserted throughout the control flow graph. At 106, a ~~maping~~ <u>mapping</u> from each variable in the control flow graph to an assert statement may be created and initialized to empty. At 108, an assert chain search procedure may be called with entry as a parameter, wherein entry is a root node of said dominator tree. This will be described in more detail with regard to FIG. 2 and the corresponding text.

Please replace paragraph [0032] with the following rewritten paragraph:

The assert chain search procedure may take a parameter X as input. The assert chain search procedure caller 312 may contain a current variable value map of assert statements saver 314, which may save a current value in the map of assert statements for each variable. A statement traverser 316 coupled to the current variable value map of assert statements saver 314 may repeat several actions for each statement in X. A variable use finder 318 coupled to the statement traverser 316 may find each variable use in the statement. A variable use traverser 320 coupled to the statement traverser 316 and to the variable use finder 318 may repeat several actions for each variable use in the statement. An available assert statement determiner 322 coupled to the variable use traverser 320 may determine if there

Appl. No. 10/625,334
Amendment and Response dated February 16, 2007
Reply to Office Action of January 29, 2007

is an available assert statement which defines information about the corresponding variable. An assert chain creator 324 coupled to the available assert statement determiner 322 and to the variable use traverser 320 may make an assert chain from the variable use to the available assert statement. An assert statement determiner 326 coupled to the statement traverser 316 and to the assert chain creator 324 may determine if the statement is an assert statement. If it is, then a variable use mapping adder 328 coupled to the assert statement determiner 326 may add a mapping for each use in the statement to an assert statement. One of ordinary skill in the art will recognize that there are many ways to track such information, but one way to keep track would be simply to save that information. If it is determined that the statement is not an assert statement, then an assert statement reference mapping deleter 330 coupled to the assert statement determiner 326 may delete the mapping of any reference to an assert statement for each definition in the statement. Then it may be determined if there are any more statements in *X*. If so, then the process may repeat and examine the next statement. If not, the process may ~~iterative~~ iteratively call the assert chain search procedure for each child of *X* in the dominator tree using an assert chain search procedure iterative caller 332 coupled to the statement traverser 316. Then the current value of the map of assert statements for each variable may be restored using a current variable value map of assert statements restorer 334 coupled to the statement traverser 316. This may be done to restore the value of a current assert definition to where it was when the process started, because it is valid for successor basic blocks only.